Solving the Continuous Global Optimization Problem with a Quantum Algorithm

January 11, 2002

V. Protopopescu and J. Barhen

Center for Engineering Science Advanced Research
Computer Science and Mathematics Division
Computing and Computational Sciences Directorate
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6355

Abstract

We investigate the entwined roles that additional information and quantum algorithms could play in reducing the complexity of the global optimization problem (GOP). We show that: (i) a modest amount of additional information is sufficient to map the general continuous GOP into the (discrete) Grover problem; (ii) while this additional information is actually available in some GOPs, it cannot be taken advantage of within classical optimization algorithms; (iii) on the contrary, quantum algorithms offer a natural framework for the efficient use of this information resulting in a speed-up of the solution of the GOP.

1 The Global Optimization Problem

Optimization problems are ubiquitous and extremely consequential. Their theoretical and practical interest has continued to grow from the first recorded instance of Queen Dido's problem [12] to present day forays into complexity theory or large scale logistics applications (see Refs. [14], [9], [8], [6], and references therein). The formulation of the GOP is deceptively simple: find the absolute minimum (maximum) of a given function - called the objective function or functional - over the allowed range of its variables. Sometimes, the function to be optimized is not specified in analytic form and must be evaluated point-wise by a computer program, a physical device, or other construct. Such a black-box tool is called an *oracle*. Of course, the brute force approach of evaluating the function on its whole domain is either impossible - if the variables are continuous - or prohibitively expensive - if the variables are discrete, but have wide ranges, in high dimensional spaces. Since in general each oracle invocation (function evaluation) involves an expensive computational sequence, the number of function evaluations needs to be kept to a minimum. The number of invocations of the oracle measures the *query complexity* of the problem and gives a fair - although by no means unique - idea of its difficulty or "hardness" [5]. Therefore, the number of oracle invocations is one of the paramount criteria in comparing the efficiency of competing optimization algorithms.

The primary difficulty in solving the GOP stems from the fact that the familiar condition for determining extrema - namely, annulment of the gradient of the objective function - is only *necessary* (the function may have a maximim, a minimum, or not have an extremum at all !) and *local* (i.e. it does not distinguish between local and global extrema). The generic strategy to find the global minimum involves two main operations, namely: (i) descent to a local minimum, a deteministic operation, and (ii) search for the new descent region, a stochastic operation. This simple strategy is marred by severe implementation problems. First, descent assumes a certain degree of smoothness, which is not always warranted. Second, after determining a local minimum, the algorithm is usually trapped in it and special operations have to be designed to restart the search. Finally, as the dimensionality of the problem increases, the search of the phase space becomes more and more complicated and time consuming; as a result, the query complexity of the problem increases as well. The "hardness" of the GOP is well illustrated by the following example for which the approach described above seems powerless. Define the function $f : [0, 1] \to \{0, 1\}$ as follows:

$$f(x) = \begin{cases} 1 \text{ for } 0 \leq x \leq a - \epsilon/2 \\ 0 \text{ for } a - \epsilon/2 < x < a + \epsilon/2 \\ 1 \text{ for } a + \epsilon/2 \leq x \leq 1. \end{cases} \tag{1}$$

where $a \in (\epsilon/2, 1-\epsilon/2)$. Since the gradient - whenever defined - is alwyas zero, the descent phase does not make sense. To obtain the minimum of this function, one should evaluate its values within the ϵ interval around the unknown number a. If the function is defined like an oracle (i.e., if one does *not* know the position of the point a), the probability of

choosing an x within this interval is ϵ. For the n-dimensional version of this oracle, the probability becomes ϵ^n, and the complexity of the problem grows exponentially with n (the dimensionality curse). This and related issues have been deftly discussed by Wolpert and Macready in connection with their "No Free Lunch" (NFL) theorem [16].

In the light of the previous example, it would seem that without additional knowledge about (the structure of) the function there is no hope to decide upon an intelligent classical optimization strategy and one would be left with either strategies that have limited albeit efficient applicability or the tantalizing prospective of exhaustive search. However, in the last few years, a new, promising option has emerged, with quantum computing being hailed as the potential provider of efficient solutions to some of the computationally hard classical problems [10]. Indeed, Grover's [7] and Shor's [11] algorithms have provided such solutions to the problems of finding a given element in an unsorted set and the prime factorization of very large numbers, respectively. While no essentially new quantum algorithm has been proposed to date, the hope is high that, on the one hand, they would, and on the other hand, the machines needed to implement them would be available in a not too distant future.

In this Letter, we present an efficient solution to the continuous GOP, by developing a generalization of Grover's algorithm to continuous problems. This generalization requires additional information on the objective function. In many optimization problems, some of this additional information *is* actually available (see below). While other required information may be more difficult to obtain in practical applications, it is still important to understand - from a theoretical point of view - the role played by additional information in reducing the difficulty of the problem, and to be able to assess *a priori what various information is relevant and for what.* To clarify the meaning of the last phrase, assume that the objective function were an analytic function. Then, the knowledge of all its derivatives at a given point would allow, in principle, the "knowledge" of the function everywhere else in the domain of analyticity. However, to actually *find* the global minimum, the function would still have to be calculated everywhere ! In other words, the (additional) knowledge of all the derivatives at a given point cannot be *efficiently* used to *locate* the global minimum, although in principle it is equivalent to the knowledge of the function at all points. In fact, to locate the global minimum, both methods would require exhaustive - albeit different - calculations.

The remaining of this letter is structured as follows. In Section 2 we present Grover's quantum algorithm for search in unstructured sets. In Section 3, under certain assumptions, we map the continuous GOP into the discrete Grover problem, thereby solving the GOP by using Grover's algorithm. In the final Section we present some implementation considerations and discuss the relaxation of the assumptions.

2 Grover's Quantum Algorithm

A quantum computation is a sequence of unitary transformations on the initial state of the wave function, ψ. As such, quantum computation is purely deterministic and reversible.

It requires the initialization or preparation of the initial state, the actual "computation", which is the unitary evolution of the initial state along a designed trajectory in the quantum computer, and the read out of the result effected through a measurement of the final state. If the algorithm is well designed and efficient, then, with probability (much) higher than 1/2, the measurement would collapse the final state onto the desired result. Computer architectures needed to implement classical or quantum algorithm are realized in terms of gates. As opposed to classical gates that operate on bits taking values in the set $\{0, 1\}$, quantum gates operate on normalized vectors in a finite-dimensional complex Euclidian space. In principle, any quantum computer can be viewed as an assembly of elementary quantum gates, such as the NOT and CNOT gates. The NOT gate is the 2×2 matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. It acts on a *qubit*, q, which is the normalized state in a two dimensional Euclidian space, $\mathbb{C}^2$:

$$q = \alpha|0> + \beta|1>, \quad |\alpha|^2 + |\beta|^2 = 1, \tag{2}$$

by exchanging the level populations. The CNOT gate acts on four dimensional vectors in $\mathbb{C}^4$. Obviously, some of these vectors can be represented as a tensor product of two two-dimensional vectors; however other vectors in $\mathbb{C}^4$ cannot be written in this form. These latter states are called *entangled states* and play a crucial role in quantum algorithms [10]. Quantum algorithms are: (i) intrinsically parallel and (ii) yield probabilistic results. These properties reflect the facts that: (i) the wave function, ψ, is nonlocal and, in fact, ubiquituous and (ii) the quantity $|\psi|^2$ is interpreted as a probability density.

Grover's original algorithm provides a solution to the following problem. Suppose we have a set of N unsorted objects, $E = \{x_1, x_2, ...x_N\}$, and an oracle function $f : E \to \{0, 1\}$, such that $f(x_1) = 1$ and $f(x_i) = 0$, $i = 2, ...N$. Using the oracle, find the element x_1 in the unsorted set E.

Without loss of generality, we can take $N = 2^n$. On average, the classical solution of the above problem involves $\sim N/2 \sim O(N)$ evaluations of the oracle. The quantum algorithm proposed by Grover [7] reduces this number to $O(\sqrt{N})$. In a generalized version of the problem, there may be L "special" elements for which the oracle returns the value one; then the number of evaluations required to find one of them is of the order $O(\sqrt{N/L})$.

We give a brief presentation of Grover's quantum algorithm [7]. First, we identify the set E with the complex Euclidean space $\mathbb{C}^N$ and the elements $x_i \in E$ with the unit vectors in $\mathbb{C}^N$, $< x_i|x_j >= \delta_{ij}$, where δ_{ij} is the Kronecker symbol. Then construct the normalized average state of all the elements $|x_i >$:

$$|w >= \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |x_i >= \frac{1}{\sqrt{N}} |x_1 > + \frac{\sqrt{N-1}}{\sqrt{N}} |x_1^{\perp} > . \tag{3}$$

In the second representation, the unit vectors orthogonal to $|x_1 >$ are lumped together in the unit vector $|x_1^{\perp} >$, which formally reduces the problem to a bidimensional space and simplifies the presentation and interpretation of the algorithm.

We note that the scalar product $< x_1|w >= \frac{1}{\sqrt{N}} := cos\beta = sin\alpha$ where β denotes the angle between the vectors $|w >$ and $|x_1 >$ and α denotes its complement, i.e. the angle between the vectors $|w >$ and $|x_1^{\perp} >$.

The construction of the state $|w >$ can be done in $log_2 N = n$ steps by applying (in parallel) n Hadamard transformations, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ on the initial zero state, $\underbrace{|0 > \otimes ..... \otimes |0 >}_{n \text{ times}}$. In the $\{|x_1 >, |x_1^{\perp} >\}$ basis, we construct the operators:

$$I_{x_1} := I - 2|x_1 >< x_1| = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{4}$$

which executes a reflection (sign inversion) of the x_1-component of a vector and

$$I_w := I - 2|w >< w| = \begin{pmatrix} 1 - 2/N & -2\sqrt{N-1}/N \\ -2\sqrt{N-1}/N & -1 + 2/N \end{pmatrix}, \tag{5}$$

which represents a reflection (sign inversion) of the w-component of a vector. At the level of the oracle, the operator I_{x_1} is implemented by $(-1)^{f(.)}(.)$, which does not depend explicitly on the unknown element x_1, while the application of the operator I_w is obvious, since the average state is known. We define now the *amplitude amplification operator*:

$$Q := -I_w I_{x_1} = \begin{pmatrix} 1 - 2/N & 2\sqrt{N-1}/N \\ -2\sqrt{N-1}/N & 1 - 2/N \end{pmatrix} = \begin{pmatrix} \cos 2\alpha & \sin 2\alpha \\ -\sin 2\alpha & \cos 2\alpha \end{pmatrix} \tag{6}$$

which, in the compressed, two-dimensional representation of the problem, represents a rotation of the state vector with an angle 2α towards $|x_1 >$. This means that each application of the operator Q will increase the weight of the unknown vector $|x_1 >$ (which explains the name of the operator Q) and after roughly $\frac{\pi/2-\alpha}{2\alpha} \sim \frac{\pi/2-1/\sqrt{N}}{2/\sqrt{N}} \sim \frac{\pi}{4}\sqrt{N}$ applications the state vector will be essentially parallel to $|x_1 >$, whereupon a measurement of the state will yield the result $|x_1 >$ with a probability very close to unity. We mention that for (and only for) $N = 4$, the result is obtained with certainty, after only one application. In general, if one continues the application of Q, the state vector continues its rotation and the weight of $|x_1 >$ decreases; eventually, the evolution of the state is cyclic as prescribed by the unitary evolution. In the original, N-dimensional representation, the operator I_w has the representation:

$$I_w = \begin{pmatrix} 1 - 2/N & 2/N \ldots & 2/N \\ 2/N & 1 - 2/N \ldots & 2/N \\ \vdots & \ddots & \vdots \\ 2/N & \ldots & 1 - 2/N \end{pmatrix} \tag{7}$$

Using this representation, one can show explicitly that the algorithm can be implemented as a sequence of local operations such as rotations, Hadamard transforms, etc. [7]

Finally, it is easy to check that if the oracle returns the same value for all the elements, i.e. there is no "special" element in the set E, the amplification operator Q reduces to

the identity operator I and, after the required number of applications, the measurement will return any of the states with the same probability, namely $1/N$. In other words, the algorithm behaves consistently.

3 Solution of the Continuous Global Optimization Problem

Grover's algorithm has been applied previously to a discrete optimization problem, namely finding the minimum among an unsorted set of N different objects. Dürr and Hoyer [4] adapted Grover's original algorithm and solved this problem with probability strictly greater than 1/2 using $O(\sqrt{N})$ function evaluations (oracle invocations).

In this Letter, we map the *continuous* GOP to the Grover problem. Once this is achieved, one can apply Grover's algorithm and obtain an almost certain result with $O(\sqrt{N})$ function evaluations. However, the mapping of the GOP to Grover's problem cannot be done in general. It requires additional information.

Before spelling out the required information, let us revisit the "pathological" example (1). Without loss of generality, we can take $\epsilon = 1/N$ and divide the segment $[0, 1]$ into N equal intervals. By evaluating the function at the midpoint of the N intervals, we obtain a discrete function that is equal to 1 in $N - 1$ points and equal to 0 in one point, which - up to an unessential transformation - is equivalent to Grover's problem. Direct application of any Grover-like algorithm yields the corresponding result. Of course, generalization to any dimension d is trivial. Thus, a problem that seems intractable classically becomes much easier within the quantum computing framework. We shall return to this example after discussing the general case.

Consider now a real function of d variables, $f(x_1, x_2, ..., x_d)$. Without restricting generality, we can assume that f is defined on $[0, 1]^d$ and takes values in $[0, 1]$. Assume now that: (i) there is a unique global minimum which is reached at zero; (ii) there are no local minima whose value is infinitesimally close to zero; in other words, the values of the other minima are larger than a constant $\delta > 0$, and (iii) the size of the basin of attraction for the global minimum measured at height δ is known; we shall denote it Δ.

Then our implementation paradigm is the following: (i) instead of $f(.)$, consider the transformation $g(.) := (f(.))^{1/m}$. For sufficiently large m, this function takes values very close to one, except in the vicinity of the global minimum, which maintains its original value, namely zero. Of course, other transformations can be used to achieve essentially the same result. We calculate m such that $\delta^{1/m} = 1/2$. To avoid technical complications that would not change the tenure and conclusions of the argument, we assume that $\Delta = 1/M$ where M is a natural number, and divide the hypercube $[0, 1]^d$ in small d-dimensional hypercubes with sides Δ. At the midpoint of each of these hypercubes, we define the function $h(x) := int[g(.) + 1/2]$ (here int denotes the integer part). The function $h(.)$ is defined on a discrete set of N points, $N = M^d$, and takes only values one and zero; by the hypotheses on the function f and our choice of constants, the region on which h takes the

value zero is a hypercube with side Δ. Thus we have reduced the problem to the Grover setting. Application of Grover's algorithm to the function $h(.)$, will result in a point that returns the value zero; by construction, this point belongs to the basin of attraction of the global minimum. We return then to the original function $f(.)$ and apply the descent technique of choice that will lead to the global minimum. If the basin of attraction of the global minimum is narrow, the gradients of the function $f(.)$ may reach very large values which may cause overshots. Once that phase of the algorithm is reached, one can apply a scaling (dilation) transformation that maintains the descent mode but moderates the gradients. On the other hand, as one approaches the global minimum, the gradients may become very small and certain acceleration techniques based on non-Lipschitzian dynamics may be required [1, 2]. If the global minimum is attained at the boundary of the domain, the algorithm above will find it without additional complications.

4 Practical Implementation Considerations

Conditions (i)-(iii) imposed on the functions $f(.)$ are rather strong, *sufficient* conditions, that we have considered mainly for convenience. We note however that: (a) these conditions *are* both satisfied and *explicitly given* for the academic "golf course" example mentioned before; (b) while they do not help reduce the complexity of the classical descent/search algorithm, they make a remarkable difference in the quantum framework; and (c) they can be replaced by weaker conditions.

Condition (i) is in fact satisfied by a large class of important practical problems, namely parameter identification encountered e.g. in remote sensing, pattern recognition, and, in general, inverse problems. In these problems the absolute minimum, namely zero, is attained for the correct values of the parameters, matching of patterns, and fitting of output to input. Condition (i) can be relaxed in the sense that the function may have multiple global minima, all equal to zero. Functions with multiple global minima will simply result in Grover problems with multiple "special" elements and can be treated accordingly if the number of global minima is known.

Condition (ii) can be replaced with the much more reasonable assumption that f has a finite number of local minima. This would prevent the value of any local minimum to be infinitesimally close to the value of the global minimum.

Condition (iii) is the most difficult to fulfill in practical problems. However, this assumption could also be relaxed with no significant performance loss if more efficient (e.g. exponentially fast) unstructured search quasntum algorithms were available. Recently, Chen and Diao [3], proposed such an algorithm which was supposed to achieve an exponential (as opposed to polynomial) speed-up of the unstructured search. Unfortunately, substantial complexities subtly hidden in one of the steps of the algorithm renders it even less efficient than Grover's [13].

Strong as it may seem, condition (iii) is essentially reasonable, since it roughly requires that the basin of attraction of the global minimum of the function f be sufficiently narrow, but not too narrow. Indeed, if the basin is narrow, condition (iii) would discourage

unproductive exhaustive search and warrant the application of a parsing algorithm, like Grover's. On the other hand, if the basin were infinitesimally narrow, then any algorithm, no matter how fast, would still require a number of evaluations in excess of any prescribed bound. The bottom line is that without any additional information at all, there could be no *general* efficient algorithm for the GOP [15].

Despite their scarcity and still very elusive implementation prospects, quantum algorithms could bring very efficient solutions and eventually practical implementations to hard computational problems. It seems safe to assume that - like the two algorithms proposed so far [11], [7] - future quantum algorithms will be much more "problem tailored" than their classical counterparts. Therefore, specific additional information is both needed and expected. While this information may be difficult to obtain, its role is crucial and its benefits could significantly outweigh its cost.

5 Acknowledgments

This work was partially supported by the Material Sciences and Engineering Division Program of the DOE Office of Science under contract DE-AC05-00OR22725 with UT-Battelle, LLC. We thank Drs. Robert Price and Iran Thomas from DOE for their support. V. P. thanks Dr. Cassius D'Helon for an enlightening discussion on Chen and Diao's algorithm and for a careful reading of the manuscript.

References

[1] J. Barhen and V. Protopopescu, Generalized TRUST Algorithm for Global Optimization, *State of the Art in Global Optimization*, C.A. Floudas and P.M. Pardalos eds., pp. 163-180, Kluwer Academic Press, Dordrecht, Boston 1996.

[2] J. Barhen, V. Protopopescu, and D. Reister, TRUST: A Deterministic Algorithm for Global Optimization, *Science* **276**, 1094-1097 (1997).

[3] G. Chen and Z. Diao, Exponentially Fast Quantum Search Algorithm, quant-ph/0011109.

[4] C. Dürr and P. Hoyer, A Quantum Algorithm for Finding the Minimum, quant-ph/9607014

[5] H. Lydia Deng and J. A. Scales, Characterizing Complexity in Generic Optimization Problems, Preprint, center of Wave phenomena, Golden, Colorado, CWP-208P, October 1996.

[6] C. A. Floudas and P. M. Pardalos, eds. *State of the Art in Global Optimization: Computational Methods and Applications*, Kluwer Academic Publishers, Dordrecht, Boston, 1996.

[7] L. K. Grover, Quantum Mechanics Helps in Searching a Needle in a Haystack, *Phys. Rev. Lett* **78**, 325-328 (1997).

[8] W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds. *Large Scale Optimization: State of the Art*, Kluwer Academic Publishers, Dordrecht, Boston, 1994.

[9] R. Horst and H. Tuy, *Global Optimization*, 2d ed., Springer-Verlag, Berlin, 1993.

[10] M. Nielsen and I. Chuang, *Quantm Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000; A. O. Pittenger, *An Intyroduction to Quantum Computing Algorithms*, Birkhäuser, Boston, 2000.

[11] P. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994, p. 124-134.

[12] D. R. Smith, *Variational Methods in Optimization*, Prentice Hall, Inc., Englewood Cliffs, N.J., 1974.

[13] C. C. Tu and G. L. Long, Chen and Diao's Quantum Search Algorithm is Not Exponentially Fast, quant-ph/0110098.

[14] A. Törn and A. Zilinskas, *Global Optimization*, Springer-Verlag, Berlin, 1989.

[15] J. F. Traub, G. W. Wasilkowski, and H. Wozniakowski, *Information-Based Complexity*, Academic Press, Boston, 1988.

[16] D. H. Wolpert and W. G. Macready, No Free Lunch Theorems for Optimization, *IEEE Trans. on Evolutionary Computing*, **1** 67–82 (1997); see also W. G. Macready and D. H. Wolpert, What Makes an Optimization Problem Hard ?, *Complexity*, **5**, 40–46 (1996).